MISSION SAFETY CRITICAL SYSTEMS

RESEARCH & APPLICATIONS

ricis

SYMPOSIUM '92

PERFORMANCE COMPARISON OF TOKEN RING NETWORKS FOR REAL-TIME APPLICATIONS

Sanjay Kamat and Wei Zhao

ABSTRACT

The ability to guarantee the deadlines of synchronous messages while maintaining a good aggregate throughput is an important consideration in the design of distributed real-time systems. In this paper, we study two token ring protocols, the priority driven protocol and the timed token protocol; for their suitability for hard real-time systems. Both these protocols use a token to control the access to the transmission medium. In a priority driven protocol, messages are assigned priorities and protocol ensures that messages are transmitted in the order of their priorities. Timed token protocol does not provide for priority arbitration but ensures that the maximum access delay for a station is bounded.

For both the protocols, we first derive the *schedulability conditions* under which the transmission deadlines of a given set of synchronous messages can be guaranteed. Subsequently we use these schedulability conditions to quantitatively compare the *average case behavior* of these protocols. This comparison demonstrates that each of these protocols has its domain of superior performance and neither dominates the other for the entire range of operating conditions.

BIOGRAPHY

Wei Zhao is currently an Associate Professor at Texas A & M University. He has published extensively in the areas of scheduling algorithms, communications protocols, distributed real-time systems, concurrence control in database systems, and resource management in operating systems. He received the Best Paper Award in the IEEE International Conference on Distributed Computing Systems for a paper on hard real-time communications.

Sanjay Kamat received B.Tech (1985) and M.Tech. (1987) degrees from Indian Institute of Technology, Bombay, India. Currently, he is a Ph.D. candidate in the Department of Computer Science at Texas A & M University. His research interests include distributed systems, real-time systems, and computer networks.

Performance Comparison of Token Ring Protocols for Hard-Real-Time Communication

Sanjay Kamat and Wei Zhao

Department of Computer Science
Texas A&M University
College Station, TX

Presentation Outline

- Introduction
- System Model
- Priority Driven Protocol
- Timed Token Protocol
- Comparison Results
- Conclusions

1. Introduction

- Objective -

To evaluate the performance of two token ring protocols

- The Priority Driven Protocol (eg. IEEE 802.5)
- The Timed Token Protocol (eg. FDDI)

for real-time applications.

Introduction (Cont'd)

Key performance issues in real-time networks

- Guaranty
- Predictability

Introduction (Cont'd)

Three questions to be answered:

1. Given a set of real-time messages and network parameters, will all the messages always meet their deadlines?

   • Schedulability Conditions

2. On average, how high can the load be, before a protocol breaks down?

   • Average Breakdown Utilization

3. Does one protocol perform better than the other? Under what conditions?

2. System Model

- Network Model
  - * Ring Network with number of nodes = n
  - * BW - Bandwidth
  - * θ - Token walk time (round trip delay)

- Message Model
  - * All real-time messages are synchronous (periodic).
  - * One synchronous message stream per node.
    - C_i - Payload message transmission time
    - C'_i - Augmented message transmission time (includes overheads)
    - P_i - Message period
  - * Deadline = end of message period
  - * Utilization

$$U = \sum C_i / P_i$$

3. Priority Driven Protocol

- Basic Protocol Description (IEEE 802.5)

  * Token regulates access to ring.
  * Messages assigned priorities.
  * Messages divided into frames.
  * Token and message frame headers have two priority fields:
    - service priority and reservation priority fields.
  * Node captures a token if it has messages with priority higher than that of the token.
  * Other nodes claim the next token via reservation field.
  * Token holding timer controls maximum number of frames transmitted by a node.
  * Transmitting node releases a new token with appropriate priority.

Priority Driven Protocol for Real-Time Applications

- The Objective :
  To guarantee the deadlines of synchronous messages.

- Problem analogous to processor scheduling for periodic tasks.

- Real-Time scheduling theory.
  Rate Monotonic Scheduling (RMS) Algorithm (Liu and Layland)

  * Optimal static priority algorithm.
  * Assigns priorities to tasks in inverse relation to periods.
  * Requires preemption.
  * Worst Case Achievable Utilization.
    (Minimum Breakdown Utilization) is 69%.
  * Average Breakdown Utilization is approximately 88%.
    (Lehoczky, Sha and Ding)

Implementing RMS on a token ring

Proposed by Strosnider and Marchok.

- Assign priorities to synchronous messages according to the Rate Monotonic rule.

- Token Holding Timer at each node set to allow at most one frame transmission.

{ Leads to priority arbitration at frame level.}
{ Provides an approximate implementation of preemption}

- Choice of frame size
  - a trade-off between enhanced responsiveness and frame transmission overheads.

Schedulability Criteria for RMS

- Exact schedulability conditions derived by Lehoczky, Sha and Ding for cpu scheduling using RMS.

- Basic idea - for each task, the total demand for resource time should be less than or equal to the available time.

- n Periodic tasks can be scheduled by the rate monotonic algorithm for all task phasings if

$$\boxed{\forall i,\ 1 \le i \le n,\ \min_{(k,l) \in R_i} \left(\sum_{j=1}^{i-1} \frac{C_j}{l P_k} \left\lceil \frac{l P_k}{P_j} \right\rceil + \frac{B_i}{l P_k} \right) \le 1}$$

where $R_i = \{ (k,l) \mid 1 \le k \le i,\ l = 1, \ldots, \lfloor P_i / P_k \rfloor \}$

(Sha, Rajkumar, Lehoczky)

B_i is the worst case **blocking time** of task i.

Extending the schedulability conditions to token ring

We need to do the following

- Compute the augmented message transmission times by accounting for overheads.
  - Overheads associated with frame transmission.
  - Token circulation overheads.
- Compute the worst case blocking time for a message.

Priority Driven Protocol (Cont'd)

- Computing Augmented Message Transmission Times (C'_i)

  * Message divided into frames
    Frame transmission time $F = F_{info} + F_{ovhd}$

  * $K_i = \lceil C_i / F_{info} \rceil$ total number of frames

    $L_i = \lfloor C_i / F_{info} \rfloor$ number of full length frames

  * Effective frame transmission time
    If $\Theta \geq F$, effective frame transmission time $= \Theta$
    { Since transmitting node has to wait for header to return}

    Otherwise, it is F
    { at least for all but the last frame }

Priority Driven Protocol (Cont'd)

The final expressions for augmented message transmission time

- Implementation using IEEE 802.5 standard protocol.

$$C'_i = \begin{cases} K_i * \Theta + K_i * \Theta / 2 & \text{when } F \leq \Theta \\ L_i * F + (K_i - L_i) * \max(C_i - L_i * F, \Theta) + K_i * \Theta / 2 & \text{otherwise} \end{cases}$$

Priority Driven Protocol (Contd)

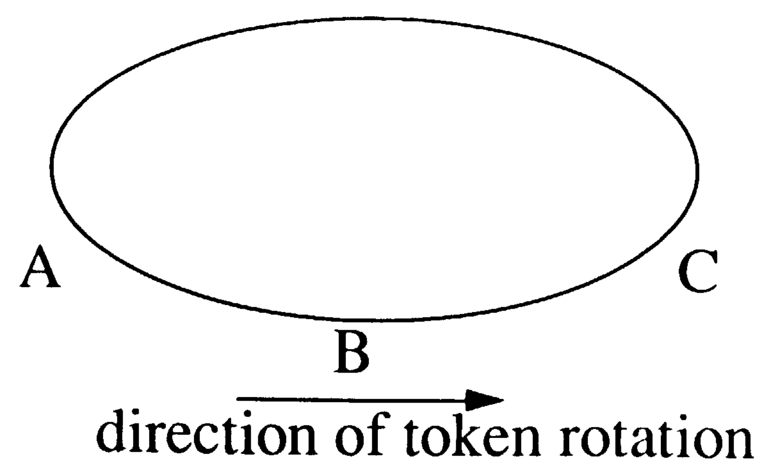
Blocking : Priority inversion due to

- Insufficient priority levels (ignored in this study)
- Approximate nature of preemption
- Bad arrival phasing

Remark:

The worst case blocking interval B_i for any message is $2 * \max(F, \Theta)$.

Example



Schedulability Conditions for Priority Driven Protocol

- Replace task computation times
  by augmented message transmission times.

- Use worst case message blocking time $B = 2 * \max(F,\Theta)$ in place of B_i.

- n Periodic messages can be scheduled by the priority driven protocol (RMS implementation) for all task phasings if

$$\boxed{\forall i,\ 1 \le i \le n,\ \min_{(k,l) \in R_i} \left\{ \sum_{j=1}^{i-1} \frac{C'_j}{lP_k} \left\lceil \frac{lP_k}{P_j} \right\rceil + \frac{B}{lP_k} \right\} \le 1}$$

where $R_i = \{ (k,l) \mid 1 \le k \le i,\ l = 1,\ldots,\lfloor P_i/P_k \rfloor \}$

4. Timed Token Protocol

- Amount of time that elapses between channel access times is bounded.
- This bound can be used to calculate the worst case proportion of time available to a node to transmit messages, and hence to guarantee message deadlines.
- Priorityless token is circulated, resulting in a round robin scheduling of transmission.

The Timed Token Protocol (Cont'd .1)

- TTRT (Target Token Rotation Time) gives expected token rotation time.
- $\tau = \Theta$ + Protocol overheads ... Denotes the portion of a token rotation that may not be available for synchronous message transmission
- Each node is allocated a portion of TTRT - τ, known as its synchronous capacity (denoted as Hi).
- Hi gives maximum amount of time node i can send synchronous messages each time it receives the token.
- Asynchronous messages can only be sent if actual token rotation time was less than TTRT.

Timed Token Protocol (Cont'd .2)

Guaranteeing messages depends on appropriately allocating the synchronous capacities, Hi.

If Hi is too small, a node may not have enough time to send its synchronous messages.

If Hi is too large, the token rotation time may become too large, e.g., larger than a message period.

Synchronous Capacity Allocation Schemes

- Full length allocation scheme:

  $Hi = C'i$

- Equal partition, usable portion of TTRT is divided equally among the nodes:

  $Hi = (TTRT - \tau)/n$

- Proportional allocation scheme:

  $Hi = (TTRT - \tau) * C'i/Pi$

- Normalized proportional allocation scheme:

  $Hi = ((TTRT - \tau) * C'i/Pi)/U$

Synchronous Capacity Allocation Schemes (Contd)

- Algorithm to generate optimal Hi's has been found (Chen et. al)
- Local capacity allocation scheme (Gopal et. al.)

$$Hi = C'_i / (q_i - 1)$$

where

$$q_i = \lfloor P_i / TTRT \rfloor$$

and

$$C'_i = C_i + \lceil C'_i / Hi \rceil * F_{ovhd}$$

All these schemes take TTRT as an input parameter.

Choice of TTRT

Remark : Our studies show that choice of TTRT is critical for obtaining a high real-time utilization without breakdown.

Selecting TTRT as $\sqrt{\tau * P_{min}}$ is found to give near optimal performance.

The local scheme is found to have a performance nearly as good as the optimal scheme for this choice of TTRT.

Schedulability Conditions for Timed Token Protocol

Any capacity allocation scheme must satisfy two constraints

- Protocol Constraint

$$\boxed{\sum Hi \leq TTRT - \tau}$$

- Deadline Constraint

  Minimum time available to transmit a message during its period $X_i \geq C'_i$

It has been shown (Chen et al) that

$$\boxed{X_i = (q_i - 1) * Hi + \max(0, \min(r_i - (\sum_{j \neq i} Hj + \tau), Hi))}$$

where $q_i = \lfloor P_i / TTRT \rfloor$ and $r_i = P_i - q_i * TTRT$

Schedulability Conditions for Timed Token Protocol (Cont'd)

Remark: For the local scheme, the deadline constraint is always satisfied.

Hence synchronous messages are guaranteed if and only if the protocol constraint is satisfied.

Hence for the local scheme, schedulability condition is

$$\boxed{\sum C_i / (q_i - 1) + n * F_{ovhd} \leq TTRT - \tau}$$

Performance Comparison

Performance Metric

- Minimum Breakdown Utilization

  { Worst case performance }

- Average Breakdown Utilization

  { Average of breakdown loads }

Average Breakdown Utilization presents a better picture of the overall performance of a protocol.

Method for Estimating Average Breakdown Utilization

Sample breakdown loads generated as follows.

- Generate initial message lengths and periods according to specified distribution.
- Uniformly scale up the message lengths till breakdown condition is reached.

5. Comparison

n = number of nodes = 100

d = distance between neibhouring nodes = 100 meters

Average bit delay per station -
4 for priority driven protocol
75 for timed token protocol

Message periods generated using a uniform distribution.
(Maximum to minimum period ratio = 10)

Number of overhead bits per frame = 112

Comparison Results

a) Packet Length = 64 Bytes
Average Period = 10 msec

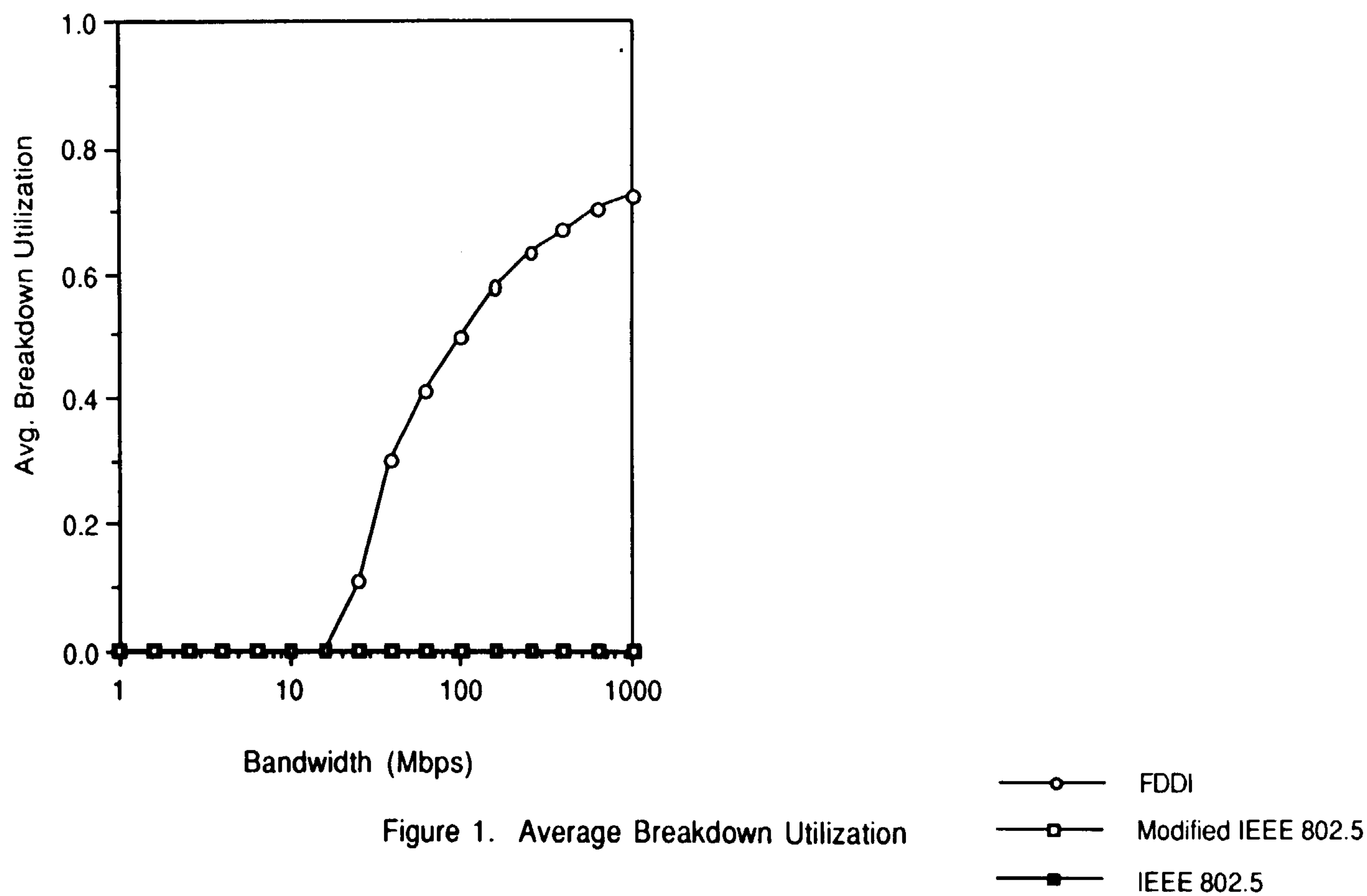


Figure 1. Average Breakdown Utilization

Comparison Results

b) Packet Length = 512 Bytes
Average Period = 10 msec

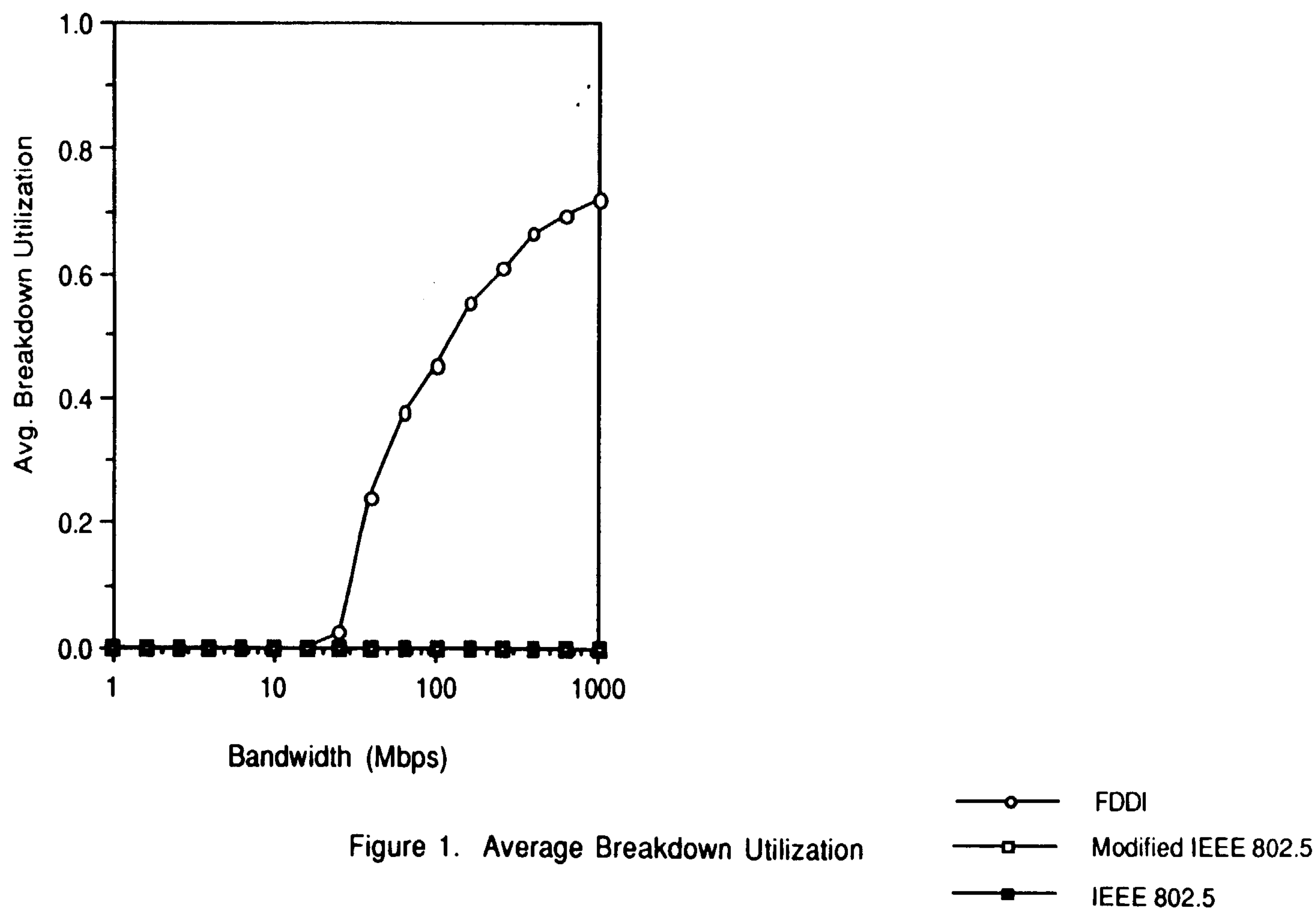


Figure 1. Average Breakdown Utilization

Comparison Results

c) Packet Length = 64 Bytes
Average Period = 100 msec

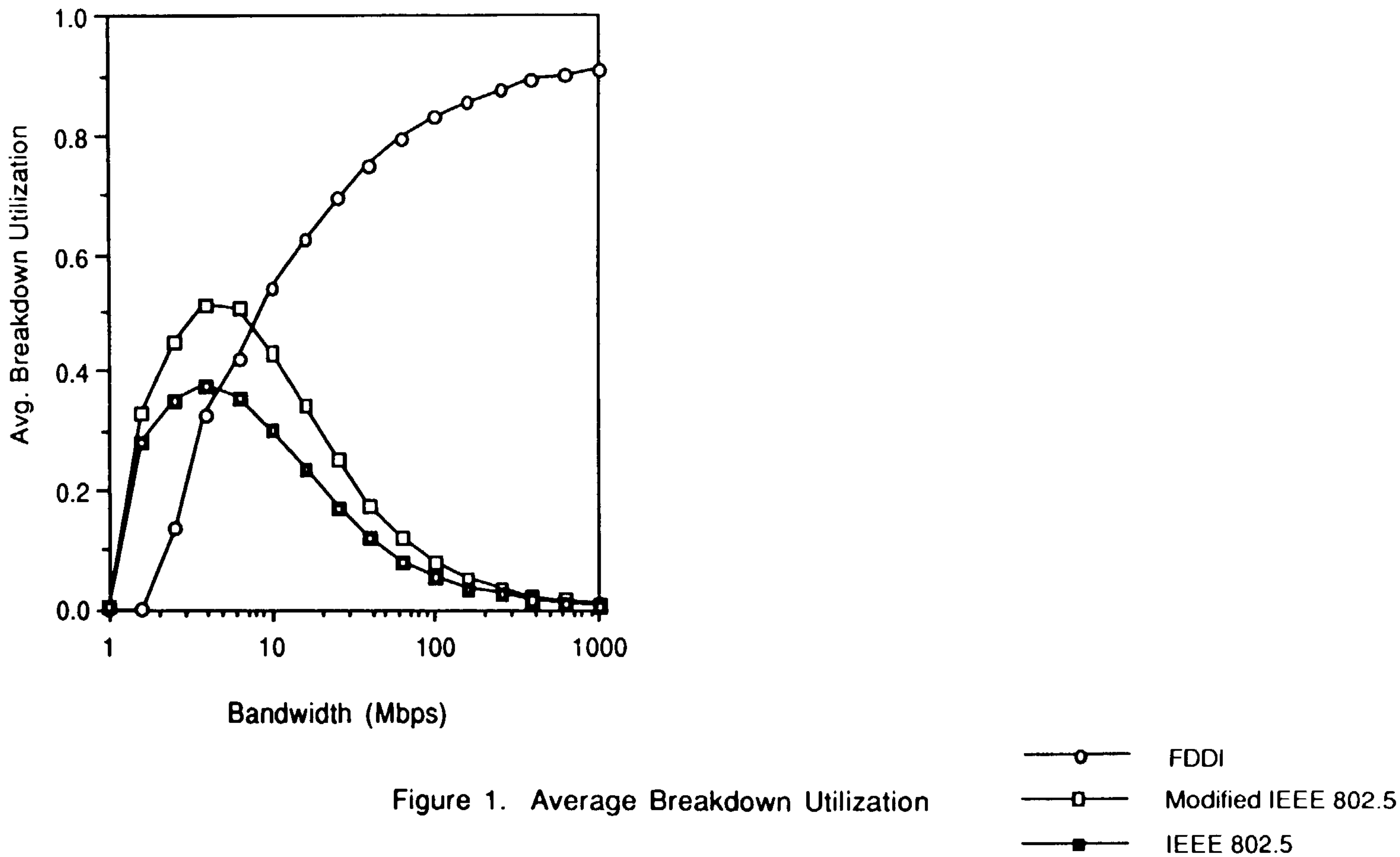


Figure 1. Average Breakdown Utilization

Comparison Results

d) Packet length = 512 Bytes
Average Period = 100 msec

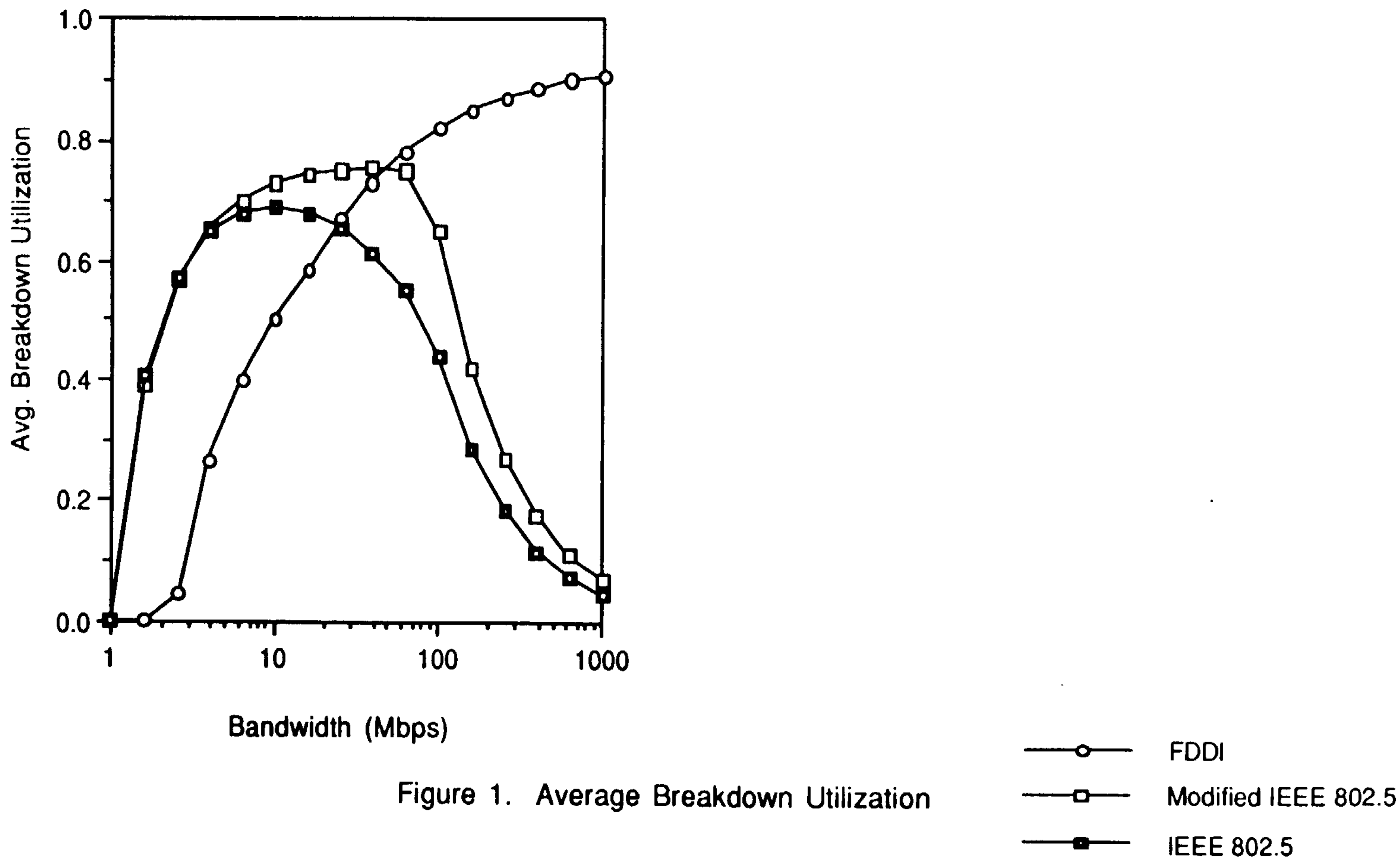


Figure 1. Average Breakdown Utilization

C-3

6. Conclusions

- Priority driven protocol and timed token protocol can be adapted for real time applications by suitable choice of protocol parameters

- Schedulability conditions aid operational level network management.

- These conditions can be used to predict the average case perforamance of protocols.

- Each protocol has its domain of superior performance.

  At low transmission speeds the priority driven protocol works better as it efficiently implements optimal scheduling strategy.

  At high bandwidths, as the priority arbitration overheads dominate,the timed token protocol works better.